KDiff3 - A File And Directory Comparison And Merge Tool

Author: Joachim Eibl, 2003 (joachim.eibl@gmx.de)

Homepage: http://kdiff3.sourceforge.net

Textcomparison

```
#include "stdio.h"                      #include <stdio.h>

int readFile(char* fileName)           int readFile(char* fileName)
{                                      {
    FILE* f = fopen(fileName);             FILE* file = fopen(fileName);
                                           if( file != 0 )
```

KDiff3 is basically a comparison tool that shows the input files side by side, and there a many such tools. But most other tools don't offer much more than a graphical user-interface for the standard GNU-Diffutils. This is unfortunate because GNU-Diffutils were designed to be text-tools and to be used on a console. The output of the classic diff is made to be processed by other tools like patch. This combination is very powerful but also has its drawbacks. A graphical tool can for instance highlight the changes within a line, because if only a detail changed, this is not easy to see in the textual output of diff. Especially when only whitespace was changed, like tabulator-characters that were replaced with spaces, this is *very* hard to see. KDiff3 can highlight the exact changes, and shows where spaces or tabulator-characters were used. Still, sometimes this is annoying, so there is an option for hiding these whitespace-changes.

There also are overview-columns that show the differences at a glance. Other tools also have this. But KDiff3 also distinguishes between "important" changes and "not-so-important" changes like whitespace-changes. If the change in a line is only in the whitespace, then the overview-columns show a chequered bar, otherwise they show a full bar for "important" changes. There even is an option to mark changes in comments as "not-so-important". And you can completely suppress the highlighting for "not-so-important" changes.

```
                                            if( file != 0 )
                                            {
    char buf[1000];                            char buf[1000];
    fread(buf,1,1000,f);                       fread(buf,1,1000,file);

    fclose(f);                                 fclose(file);
```
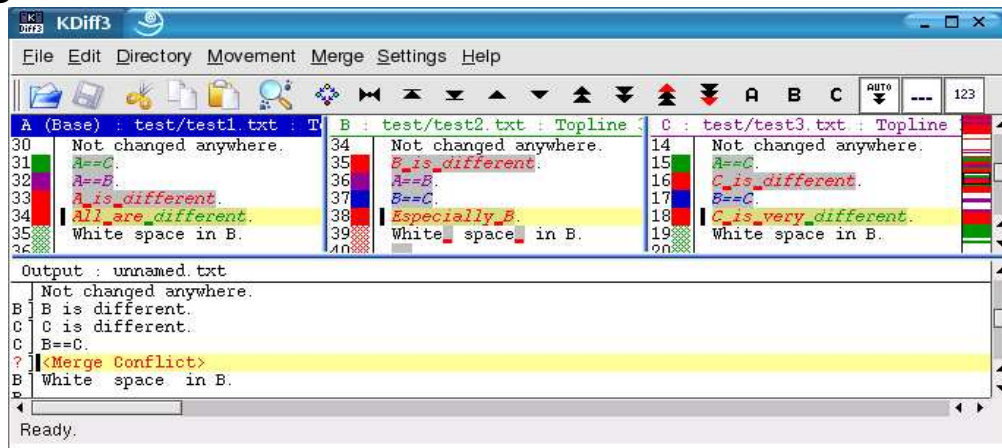
Possibly you think, that this isn't such a big thing. But at my job, I'm often asked to review the changes that other colleagues introduced. From experience I know that these details can make a huge difference then.

More options allow for special treatment of changes in numbers or the case-sensitivity during the diff.

Other features:

- line-numbers can be shown,
- searching for text,
- marking text with the mouse and copying it to the clipboard,
- pasting text from the clipboard into a diff-input window which is useful for quickly comparing small parts of a file,
- adjustable tabulator-size.

Merging Two Or Three Files

Most other graphical diff tools have merge capabilities. But usually they only allow to select either source, but few tools provide a complete editor that allows to modify the result, if a simple source selection isn't enough. KDiff3 shows the source files and for each change lets the user choose the lines from one or more inputs and allows editing of the result.

KDiff3 also supports a three-way merge. Hence the "3" in "KDiff3". The first input file is treated as the base and the other files were modified independantly (e.g. by different programmers). Knowing a common base allows KDiff3 to automatically choose the appropriate input where only one of the other inputs changed. Places that were changed in both files are conflicts that the user must solve manually.

Again the treatment of whitespace was a special concern. Due to user requests there are now possibilities to choose a certain input for whitespace changes.

Another hard issue was to get the correct input-lines lined up. But these are details and it should suffice to know that KDiff3's algorithms often get it right, when other programs have problems.
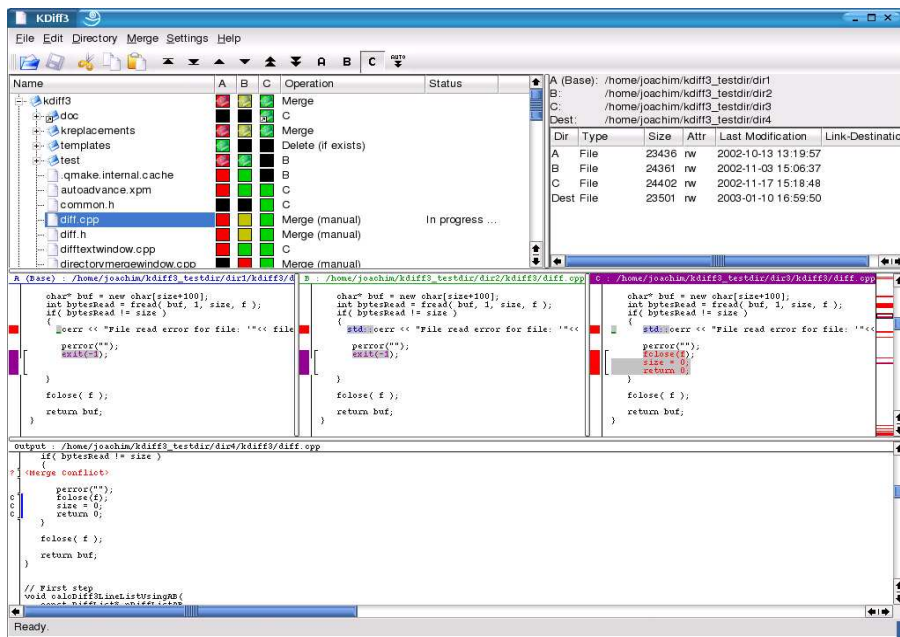
Directory Comparison, Sync And Merge

KDiff3 shows the files and directories in a list. Double-clicking a file compares all versions.

Features that are special in KDiff3:

- Support for KIO. This allows to compare local with remote directories, or data directly within a compressed archive. That's one of the coolest KDE-features.

- Options to specify file-patterns, file and directory-antipatterns, and ignoring files as CVS would via CVSIGNORE.

More interesting are the sync and merge possibilities. Like with the file merge, you can again do a complete three-way merge with one directory as base. This is a feature that most VCSs (version control systems) can do for you. But KDiff3 works independant of any version control and this allows for different uses, because sometimes using a VCS is just too much effort.

While comparing the files, KDiff3 will calculate the default operation for each file. When three input-directories were specified, then the first directory will be treated as base and if only one of the other two files was modified, then that will be the default choice. When both other files were modified, then the default operation is a manual merge. When the comparison is complete, you can modify the merge-operations manually to suit your needs, before you actually start the merge. When you start the merge, then KDiff3 will only stop and let you interact when the operation is "Manual Merge". All other operations run automatically.

When working with two directories, then you often want synchronise them. You can copy a file from one input-directory to the other directory. When merging a file, then on saving the merge-result will be saved in both directories.

Other features:

- The default operation can choose the newest file.
- Progressbars and a possibility to abort most operations.
- The merge result can be saved to another directory, so that all input directories remain unchanged.

Miscellaneous

- All features are documented.
- Because KDiff3 is a KPart, it can be used as Diff-Viewer within KDevelop, or as patch-file viewer for single files within Konqueror.
- Keyboard shortcuts for all important operations. (Most of them configurable under KDE.)
- Can be compiled even without KDE, only requiring Qt-libraries. (Runs even under Windows.)